1. Introduction

OVERFLOW 2.2 is a three-dimensional time-marching implicit Navier-Stokes code that can also operate in two-dimensional or axisymmetric mode. The code uses structured overset grid systems. Several different inviscid flux algorithms and implicit solution algorithms are included in OVERFLOW 2.2. The code has options for thin layer or full viscous terms. A wide variety of boundary conditions are also provided in the code. The code may also be used for multi-species and variable specific heat applications. Algebraic, one-equation, and two-equation turbulence models are available. Low speed preconditioning is also available for several of the inviscid flux algorithms and solution algorithms in the code. The code also supports bodies in relative motion, and includes both a six-degree-of-freedom (6-DOF) model and a grid assembly code. Collision detection and modeling is also included in OVERFLOW 2.2. The code is written to allow use of both MPI and OpenMP for parallel computing applications.

OVERFLOW 2.2 has two basic operational modes. The code can still be run in the original OVERFLOW mode, in which the OVERFLOW-D specific inputs do not have to be specified. The original OVERFLOW mode requires that all grids be supplied and assembled using PEGASUS 5 or SUGGAR prior to the start of the solution process. An example of a grid set for the original OVERFLOW mode is shown in Fig. 1.1. The OVERFLOW-D mode requires additional input files and NAMELIST input to control the DCF grid assembly, off-body grid generation, and 6-DOF or specified motion simulation. Only the near-body grids need to be supplied in OVERFLOW-D mode since Cartesian outer grids can be automatically generated prior to grid assembly using DCF. An example of a grid set for OVERFLOW-D mode is shown in Fig. 1.2.

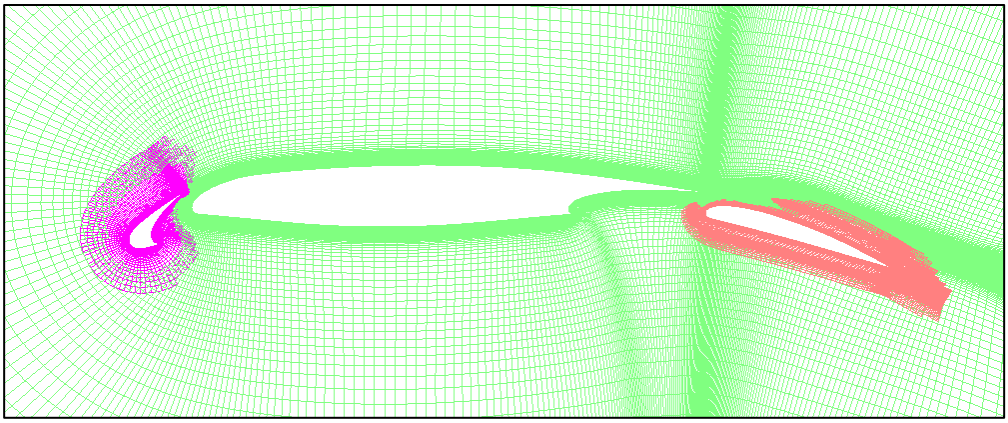

Figure 1.1. Multi-element airfoil grid for the original OVERFLOW mode. The grid assembly was performed external to OVERFLOW using PEGASUS 5.

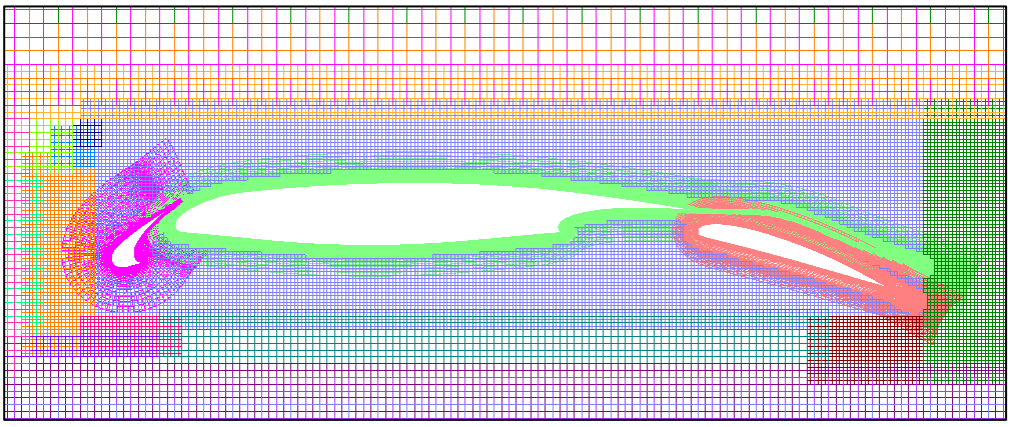

Figure 1.2. Multi-element airfoil grid for the OVERFLOW-D mode. The grid assembly and Cartesian background grid generation was performed internally using DCF.

OVERFLOW 2.2 will automatically decompose near-body and off-body grid systems to achieve load balancing for parallel runs using MPI, based on the number of processors selected for that run. The decomposed grids are automatically reassembled prior to generation of any output files, and thus the decomposition is completely transparent to the user.

This manual describes uncompressing and compiling the code, off-body grid generation, hole cutting, input files, run sequence and code execution, 6-DOF simulation, adaptation, parallel processing, and post-processing. This manual is not intended to be a comprehensive guide. Rather, it is intended as a quick reference for users already familiar with the theory used to develop the flow solver and utility codes.

1.1 Code History

The OVERFLOW 2[1,2] Navier-Stokes computational fluid dynamics (CFD) code was developed by merging the OVERFLOW[3,4] flow solver with the 6-DOF moving body capability of the OVERFLOW-D flow solver[5,6,7,8,9]. The lineage of the code is shown in Fig. 1.3. The overset grid method was developed by Benek, Buning, and Steger[10] to allow the structured grid flow solvers of the 1980's to be easily extended to more complex geometries than the grid generators would support at that time. The OVERFLOW flow solver originally incorporated the diagonal form of the implicit approximate factorization algorithm of Pulliam and Chaussee[11] and a second-order in space central difference approximation for the inviscid fluxes. Mixed second and fourth order smoothing was added to the explicit and implicit side of the equations to provide numerical stability. OVERFLOW derives its name from an acronym for "OVERset grid FLOW solver."

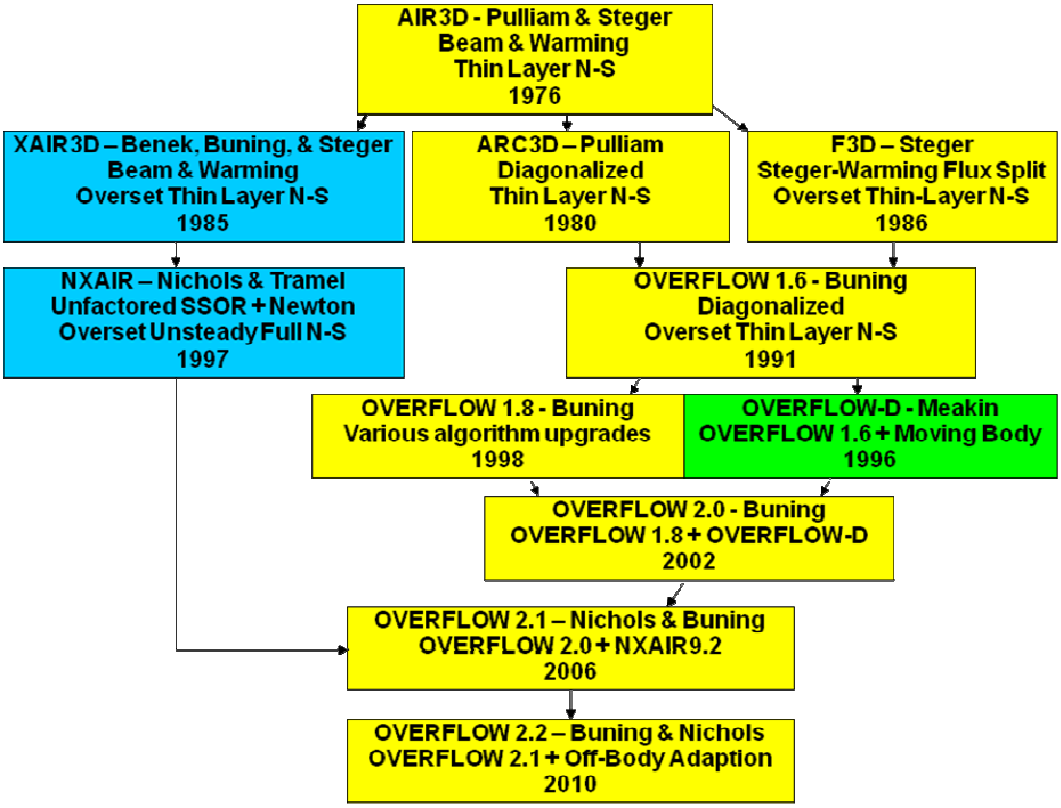

Figure 1.3. Development path for OVERFLOW 2.2.

A number of upgrades have been made to the original code including:

1. Lower Upper-Symmetric Gauss Seidel (LU-SGS) implicit solution algorithm and a Roe upwind inviscid flux scheme[12]
2. Multigrid solution procedure, low-Mach preconditioning, and a central difference/matrix dissipation inviscid flux scheme[13]

3. Parallization with OPENMP and MPI[14]
4. AUSM inviscid flux scheme[15]
5. Dual time stepping implicit solution algorithm[16]
6. HLLC[17] inviscid Riemann flux algorithm
7. Two WENO 5th order upwind algorithms[18]
8. An unfactored Successive Symmetric Over Relaxation (SSOR) implicit solution algorithm[17]
9. Three hybrid Reynolds Averaged Navier-Stokes/Large Eddy Simulation (RANS/LES) turbulence models[17]
10. Wall functions for the transport equation turbulence models[17]
11. An HLLC and WENO based species transport equation capabiilty[17]

1.2 OVERFLOW 2.2 Capabilities

The following is a summary of the capabilities currently included in OVERFLOW 2.2:

Dimensions:
- 2D
- Axisymmetric
- 3D

Flow equation sets:
- Compressible Euler
- Compressible thin-layer Navier Stokes
- Compressible full Navier Stokes
- Low Mach preconditioning

Inviscid flux algorithms:
- 2nd/4th/6th-order central difference with smoothing
- Yee Symmetric TVD
- AUSM+ upwind
- 3rd-order Roe upwind
- 3rd-order HLLC upwind
- 5th-order WENO and WENOM upwind
- Low-Mach preconditioning available for central difference, Roe, HLLC, WENO, and WENOM

Flux limiters for upwind schemes:
- Koren
- Minmod
- van Albada

Smoothers for central difference schemes:
- F3D dissipation scheme
- ARC3D dissipation scheme
- TLNS3D dissipation scheme
- Matrix dissipation scheme

Implicit solvers:
- ADI Beam-Warming block tridiagonal solver with either central difference or upwind Steger-Warming flux Jacobians
- Steger-Warming 2-factor scheme
- ADI Pulliam-Chaussee scalar pentadiagonal solver
- LU-SGS solver
- D3ADI diagonalized solver
- SSOR solver
- Low-Mach preconditioning for Beam-Warming, Pulliam-Chaussee, D3ADI, and SSOR solvers

Turbulence models:
- Baldwin-Lomax algebraic model with wake model
- Baldwin-Barth 1-equation transport model
- Spalart-Allmaras 1-equation transport model
- Spalart-Allmaras detached eddy simulation (DES) and delayed DES (DDES) hybrid RANS/LES models

- k-ω 2-equation transport model
- SST 2-equation transport model
- SST DES and SST DDES 2-equation transport hybrid RANS/LES models
- SST Multi-Scale (MS) 2-equation transport hybrid RANS/LES model
- Wall functions for all of the transport turbulence models
- Rotation and curvature corrections for 1- and 2-equation models
- Temperature correction for 2-equation models

Boundary conditions:
- Slip wall
- Adiabatic no-slip wall
- Constant temperature no-slip wall
- Symmetry plane
- Polar axis
- Periodic
- Riemann far-field
- Constant total pressure and temperature inflow
- Constant pressure outflow
- Extrapolation outflow
- Specified mass flow outflow
- Frozen
- Overset interpolated

Gas models:
- Perfect gas
- Variable γ(mixture of perfect gases)
- Multi-species

Moving body:
- Prescribed motion
- Six-degree-of-freedom (6-DOF) motion
- Collision detection and simulation

Grid generation, assembly, and adaptation
- Automatic generation of Cartesian off-body grids
- Grid assembly using DCF
- Adaptation of Cartesian off-body grids to flow variables or body motion

Code performance:
- MPI coarse-grained parallelism
- OpenMP fine-grained parallelism
- Automatic grid decomposition for load balancing

1.3 Navier-Stokes Equation Implicit Solution Procedure

OVERFLOW 2.2 solves the Navier-Stokes in generalized coordinates. The Navier-Stokes equations may be written as

$$\frac{\partial \vec{q}}{\partial t} + \frac{\partial \vec{E}}{\partial \xi} + \frac{\partial \vec{F}}{\partial \eta} + \frac{\partial \vec{G}}{\partial \zeta} = 0 \tag{1.1}$$

where q is the vector of conserved variables

$$\vec{q} = V\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_0 \end{bmatrix} \tag{1.2}$$

The linearized Euler implicit form of Eq. (1.1) including sub-iterations is given by

$$\begin{aligned} &\left[I + \frac{\Delta\tau}{S_D}\left(\partial_\xi A + \partial_\eta B + \partial_\zeta C\right)\right]\Delta q^{n+1,m+1} = \\ &-\left[\frac{(1+\theta)\Delta\tau}{S_D \Delta t}\left(q^{n+1,m} - q^n\right) - \frac{\theta\Delta\tau}{S_D\Delta t}\Delta q^n + \frac{\Delta\tau}{S_D} RHS^{n+1,m} \right] \end{aligned} \tag{1.3}$$

Here θ=0 for first order time differencing, and θ=½ for second order time differencing. An artificial time term $\left(\frac{\Delta t}{(1+\theta)\Delta\tau}\right)$ has been explicitly added for dual time stepping. This term is not included when using Newton subiterations. The pseudo time ($\Delta\tau$) may vary throughout the flow field when a local time step is employed. The artificial time term must converge at each physical time step (i.e., $\Delta q^{n+1,m+1} = 0$) to assure time accuracy. The S_D term is defined as $1 + \frac{(1+\theta)\Delta\tau}{\Delta t}$ for dual time stepping and as $(1+\theta)$ for Newton subiterations. The two time steps are equal to each other and do not vary in the field for the Newton subiteration ($\Delta t = \Delta\tau$). The explicit viscous and inviscid fluxes are included in the term *RHS* given by

$$RHS = \frac{\partial \vec{E}}{\partial \xi} + \frac{\partial \vec{F}}{\partial \eta} + \frac{\partial \vec{G}}{\partial \zeta} \tag{1.4}$$

Eq. (1.3) has the general matrix form $Ax=b$. The first bracketed term in Eq. (1.3) is the left hand side matrix A. The second bracketed term in Eq. (1.3) represents the vector b. $\Delta q^{n+1,m+1} = q^{n+1,m+1} - q^{n+1,m}$ contains the change in the solution vector at the latest time step ($n+1$) and sub-iteration ($m+1$) when sub-iterations are employed in the solution process. $\Delta q^{n+1,1} = q^{n+1,1} - q^{n,1}$ contains the change in the solution vector at the latest time step ($n+1$) if no sub-iterations are used. If the time step remains constant everywhere in the field then Eq. (1.3) represents a Newton sub-iteration ($\Delta t = \Delta\tau$). If the pseudo time step is allowed to vary throughout the field then Eq. (1.3) represents a dual time stepping algorithm. In both cases the sub-iteration is used to improve the accuracy of the solution at each global time step. The individual grids are solved implicitly, but the overset interpolated boundaries are updated explicitly at each sub-iteration. All of the physical boundary conditions with the exception of the periodic boundaries are also treated explicitly. Hence the sub-iterations improve the global convergence at each time step by allowing a global exchange of information among the grids and improving the accuracy of the physical boundaries.

Solving the above system of discreet equations requires inversion of the A matrix. Direct inversion of the matrix A for three dimensional flows requires a large amount of computational time and memory. Various approximations have been made to expedite the procedure in the past. The Newton subiteration form of Eq. (1.3) can be factored[19] in space

$$\begin{aligned} &\left[I + \frac{\Delta t}{1+\theta}\partial_\xi A\right]\left[I + \frac{\Delta t}{1+\theta}\partial_\eta B\right]\left[I + \frac{\Delta t}{1+\theta}\partial_\zeta C\right]\Delta q^{n+1,m+1} = \\ &-\left[\left(q^{n+1,m} - q^n\right) - \frac{\theta}{1+\theta}\Delta q^n + \frac{\Delta t}{1+\theta} RHS^{n+1,m} \right] + Error \end{aligned} \tag{1.5}$$

where the factorization error (*Error*) is given by

$$Error = \left[\left(\frac{\Delta t}{1+\theta} \right)^2 \left(\partial_\xi A \partial_\eta B + \partial_\xi A \partial_\zeta C + \partial_\eta B \partial_\zeta C \right) + \left(\frac{\Delta t}{1+\theta} \right)^3 \left(\partial_\xi A \partial_\eta B \partial_\zeta C \right) \right] \Delta q^{n+1,m+1} \tag{1.6}$$

The factorization error term in Eq. (1.5) is usually ignored, resulting in an approximate factorization of Eq. (1.3). The factorization error is scaled by the time step squared and cubed for three dimensional calculations. The factorization error is only scaled by the time step squared for two dimensional and axisymmetric calculations since $\partial_\zeta C = 0$. The factorization error can limit or prevent convergence for large time steps.

The approximate factorization in Eq. (1.5) is called a three factor alternating direction implicit (ADI) scheme. *A*, *B*, and *C* are block tridiagonal matrices for structured grids with central difference or first-order spatial upwind implicit flux Jacobians. The factored system can be solved efficiently by inverting the block tridiagonal matrices in each direction.

The *A*, *B*, and *C* matrices in Eq. (1.5) may be decomposed into eigenvalues (Λ) and eigenvectors (*X*) as

$$\begin{aligned} A &= X_A \Lambda_A X_A^{-1} \\ B &= X_B \Lambda_B X_B^{-1} \\ C &= X_C \Lambda_C X_C^{-1} \end{aligned} \tag{1.7}$$

Pulliam and Chaussee[11] suggested pulling out the eigenvector matrices from Eq. (1.5), producing the following system of equations

$$\begin{aligned} &X_A \left[I + \frac{\Delta t}{1+\theta} \partial_\xi \Lambda_A \right] X_A^{-1} X_B \left[1 + \frac{\Delta t}{1+\theta} \partial_\eta \Lambda_B \right] X_B^{-1} X_C \left[I + \frac{\Delta t}{1+\theta} \partial_\zeta \Lambda_C \right] X_C^{-1} \Delta q^{n+1,m+1} = \\ &- \left[\left(q^{n+1,m} - q^n \right) - \frac{\theta}{1+\theta} \Delta q^n + \frac{\Delta t}{1+\theta} RHS^{n+1,m} \right] + Error \end{aligned} \tag{1.8}$$

Eq. (1.8) results in a scalar pentadiagonal matrix form in each factored direction when mixed second and fourth order smoothing is included on the implicit side of the equation. The approximations used to derive Eq. (1.8) do affect time accuracy for CFL numbers greater than one. The inversion of a scalar pentadiagonal matrix at each point can be done very efficiently, making this algorithm extremely fast in terms of time/iteration/point. The diagonal schemes implemented in OVERFLOW 2.2 have made the code one the fastest available for obtaining steady state solutions.

The Pulliam-Chaussee[11], Beam-Warming[19], and the diagonally-dominant ADI (DDADI)[20] implicit algorithms in OVERFLOW 2.2 are three-factor ADI schemes. The F3D[21] algorithm is a partially flux-split two-factor ADI scheme that uses Steger-Warming flux vector splitting. The LU-SGS[10] algorithm in OVERFLOW 2.2 is also a two-factor scheme. Dual time stepping and local time stepping methods have been implemented to improve convergence and stability for these algorithms. For many unsteady applications, the time step required for numerical stability is too small to be practical. The three-factor ADI schemes have been the preferred solution algorithm in the past because of their low memory requirements and because they are relatively fast in terms of time/point/iteration. For a grid of dimensions (jd, kd, ld) the memory required to store the flux Jacobian matrices during the solution process is 19*max(jd*kd, jd*ld, kd*ld) for the scalar pentadiagonal algorithm and 80*max(jd*kd, jd*ld, kd*ld) for the Beam-Warming[19] algorithm since the matrices may be inverted a plane at a time in each computational direction.

Several flow solvers have been developed to solve the unfactored system of equations (Eq. (1.3)) using relaxation procedures. This approach eliminates the factorization error at the expense of more computational work per time step and more computational memory since the entire implicit flux Jacobian matrix (*A*) must be stored for the solution process. First-order upwind Steger-Warming[21] inviscid flux Jacobians and central difference thin-layer viscous Jacobians are used in the formulation in OVERFLOW 2.2. The symmetric successive over-relaxation scheme (SSOR) used in OVERFLOW 2.2 can be written as

$$\begin{aligned} \Delta q_{j,k,l}^{mm+1} &= (1-\Omega) \Delta q_{j,k,l}^{mm} + \Omega (\overline{RHS} - \overline{A}_L \Delta q_{j-1,k,l}^{mm-1} - \overline{A}_R \Delta q_{j+1,k,l}^{mm-1} \\ &- \overline{B}_L \Delta q_{j,k-1,l}^{mk1} - \overline{B}_R \Delta q_{j,k+1,l}^{mk2} - \overline{C}_L \Delta q_{j,k,l-1}^{ml1} - \overline{C}_R \Delta q_{j,k,l+1}^{ml2}) \end{aligned} \tag{1.9}$$

The subscripts *L* and *R* denote the left and right blocks respectively of the tridiagonal matrices. The overbar indicates a pre-multiply by the inverse of the diagonal matrix A_D+B_D+C_D where the subscript *D* denotes the diagonal block of the tridiagonal matrices. The update level of Δq during the iterative matrix solution procedure is given by *mm*. The scheme uses a forward (Jacobi) sweep in *j* and symmetric (Gauss-Seidel) sweeps in *k* and *l*. For a forward sweep in *k* and *l*, the update levels are defined as

$$mk1 = mm+1,\ mk2 = mm,\ ml1 = mm+1,\ ml2 = mm \tag{1.10}$$

For a backward sweep in *k* and *l*, the update levels are defined as

$$mk1 = mm,\ mk2 = mm+1,\ ml1 = mm,\ ml2 = mm+1 \tag{1.11}$$

A symmetric sweep consists of a forward and a backward sweep. Multiple symmetric sweeps (normally 10) are performed at each sub-iteration. The relaxation parameter (Ω) is normally set to 0.9. The SSOR solver uses more memory than the factored solvers.

OVERFLOW 2.2 also includes the ability to use grid sequencing and a multigrid solution algorithm in the solution process. Grid sequencing provides an easy way to get the solution off the ground and can significantly reduce the time required to obtain a converged solution. In grid sequencing the input grid is coarsened by removing every other point for each level of sequencing used. The solution begins on the coarsest grid and moves to the successively finer grids after a specified number of iterations. This allows solutions to set up quickly. OVERFLOW 2.2 also includes a V-cycle multigrid capability. A correction term is calculated on each set of coarsened grids and applied to the residual at each time step. This allows low frequency errors to be removed from the solution rapidly and can reduce the number of time steps required to reach a steady state solution.

1.4 Low-Mach Number Preconditioning

OVERFLOW 2.2 also includes a low-Mach number preconditioning capability. At low speeds the eigenvalues of the Navier-Stokes equations

$$\lambda = \begin{bmatrix} U \\ U \\ U \\ U+c \\ U-c \end{bmatrix} \tag{1.12}$$

become widely separated and the equation set becomes stiff. Here *U* is the contravariant velocity and *c* is the local speed of sound. Preconditioning is used to scale the eigenvalues to remove this stiffness. Note that this scaling of the eigenvalues destroys the time accuracy of the equations. The low Mach number preconditioned Navier-Stokes equations can be written

$$\Gamma_p \frac{\partial \vec{q}_p}{\partial \tau} + \frac{\partial \vec{q}}{\partial t} + \frac{\partial \vec{E}}{\partial \xi} + \frac{\partial \vec{F}}{\partial \eta} + \frac{\partial \vec{G}}{\partial \zeta} = 0 \tag{1.13}$$

where Γ_p is the preconditioning matrix and $\vec{q}_p$ are the preconditioned flow variables.. The preconditioned eigenvalues using Smith-Weiss preconditioning become

$$\lambda = \begin{bmatrix} U \\ U \\ U \\ 0.5U(\beta+1) - \sqrt{0.5(\beta-1)^2 + \beta c^2} \\ 0.5U(\beta+1) + \sqrt{0.5(\beta-1)^2 + \beta c^2} \end{bmatrix} \tag{1.14}$$

where β is defined as

$$\beta = \max\left[\min\left(M^2, 1\right), \beta_{\min}\right] \tag{1.15}$$

and $\beta_{\min}$ is a user defined variable normally set to $3M_{\text{REF}}{}^2$.

1.5 Species Equations

A scalar transport equation is included in OVERFLOW 2.2 for multi-species calculations. The species equations are solved decoupled from the Navier-Stokes equations and from each other. The species equation in generalized coordinates can be written as

$$\begin{aligned}
&\frac{\partial \rho c_i}{\partial t} + \frac{\partial \rho U c_i}{\partial \xi} + \frac{\partial \rho V c_i}{\partial \eta} + \frac{\partial \rho W c_i}{\partial \zeta} = \frac{\partial}{\partial \xi}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x^2 + \xi_y^2 + \xi_z^2\right)\frac{\partial c_i}{\partial \xi}\right] + \\
&\frac{\partial}{\partial \eta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\eta_x^2 + \eta_y^2 + \eta_z^2\right)\frac{\partial c_i}{\partial \eta}\right] + \frac{\partial}{\partial \zeta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right)\frac{\partial c_i}{\partial \zeta}\right] + \\
&\frac{\partial}{\partial \xi}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z\right)\frac{\partial c_i}{\partial \eta}\right] + \frac{\partial}{\partial \xi}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z\right)\frac{\partial c_i}{\partial \zeta}\right] + \\
&\frac{\partial}{\partial \eta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z\right)\frac{\partial c_i}{\partial \xi}\right] + \frac{\partial}{\partial \eta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z\right)\frac{\partial c_i}{\partial \zeta}\right] + \\
&\frac{\partial}{\partial \zeta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z\right)\frac{\partial c_i}{\partial \xi}\right] + \frac{\partial}{\partial \zeta}\left[\left(\frac{\mu}{\sigma_L} + \frac{\mu_t}{\sigma_T}\right)\left(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z\right)\frac{\partial c_i}{\partial \eta}\right] + \\
&+ \textit{Source Terms}
\end{aligned} \tag{1.16}$$

Here c_i is the species mass fraction. The following relationships are used to obtain thermodynamic properties for each species.

$$\frac{c_p}{R} = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4 \tag{1.17}$$

$$\sum_{i=1}^{ngas} c_i = 1 \tag{1.18}$$

$$R_{mix} = \sum_{i=1}^{ngas} c_i R_i,\ c_{pmix} = \sum_{i=1}^{ngas} c_i c_{pi},\ c_{vmix} = \sum_{i=1}^{ngas} c_i c_{vi},\ \gamma_{mix} = \frac{c_{pmix}}{c_{vmix}} \tag{1.19}$$

$$\overline{T} = \frac{T}{\gamma_\infty T_\infty} = \frac{\gamma_{mix} - 1}{R_{mix}}\overline{e}_i = \frac{\overline{e}_i}{c_{vmix}} \tag{1.20}$$

1.6 Non-Dimensional Flow Variables

OVERFLOW 2.2 solves the non-dimensional Navier-Stokes, turbulence, species, and motion equations. The thermodynamic equations (based on a perfect gas assumption) used by the flow solver are

$$\begin{aligned} p &= \rho RT, && \textit{perfect gas law} \\ a^2 &= \gamma RT, && \textit{perfect gas speed of sound} \\ e_i &= c_v T \end{aligned} \tag{1.21}$$

The following definitions are used involving the specific heats c_p and c_v:

$$\begin{aligned} \gamma &= \frac{c_p}{c_v} \\ R &= c_p - c_v \\ R &= \frac{\mathcal{R}}{MW}, && MW = \textit{molecular weight} \end{aligned} \tag{1.22}$$

Alternate forms of the thermodynamic equations can be derived, such as

$$\begin{aligned} a^2 &= \frac{\gamma p}{\rho} \\ p &= (\gamma - 1)\rho e_i \end{aligned} \tag{1.23}$$

Rearrangements of the specific heat and gas constant relations can be made as well:

$$R = (\gamma - 1)c_v = \frac{(\gamma - 1)}{\gamma} c_p \tag{1.24}$$

Non-dimensionalization of flow quantities follows that of the F3D and ARC3D codes. Denoting non-dimensional quantities by an asterisk, we define

$$
\begin{aligned}
x^* &= \frac{x}{L} \\
V^* &= \frac{V}{a_\infty} \\
t^* &= \frac{ta_\infty}{L} \\
\rho^* &= \frac{\rho}{\rho_\infty} \\
p^* &= \frac{p}{\rho_\infty a_\infty^2} = \frac{p}{\gamma_\infty p_\infty} \\
T^* &= \frac{TR_\infty}{a_\infty^2} = \frac{T}{\gamma_\infty T_\infty} \\
R^* &= \frac{R}{R_\infty} \\
c_p^* &= \frac{c_p}{R_\infty} \\
c_v^* &= \frac{c_v}{R_\infty} \\
e_i^* &= \frac{e_i}{a_\infty^2} \\
\mu^* &= \frac{\mu}{\mu_\infty} \\
k^* &= \frac{k}{k_\infty} \\
\nabla^* &= L\nabla
\end{aligned}
\tag{1.25}
$$

A number of these non-dimensionalizations were driven by the desire to maintain non-dimensionalized thermodynamic equations in the same form as the dimensional equations. Substituting, we see that the following non-dimensional equations hold:

$$
\begin{aligned}
p^* &= \rho^* R^* T^* = (\gamma - 1)\rho^* e_i^* \\
a^{*2} &= \gamma R^* T^* = \frac{\gamma p^*}{\rho^*} \\
e_i^* &= c_v^* T^*
\end{aligned}
\tag{1.26}
$$

Non-dimensionalizations used for moving-body simulation parameters are given in Chapter 5.

References

1. Murphy, K.J., Buning, P.G., Pamadi, B.N., Scallion, W.I., and Jones, K.M., "Overview of Transonic to Hypersonic Stage Separation Tool Development for Multi-Stage-to-Orbit Concepts," AIAA-2004-2595, June 2004.
2. Buning, P.G., Gomez, R.J., and Scallion, W.I., "CFD Approaches for Simulation of Wing-Body Stage Separation," AIAA-2004-4838, Aug. 2004.
3. Buning, P.G., Chiu, I.T., Obayashi, S., Rizk, Y.M., and Steger, J.L., "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent," AIAA-88-4359, Aug. 1988.
4. Renze, K.J., Buning, P.G., and Rajagopalan, R.G., "A Comparative Study of Turbulence Models for Overset Grids," AIAA-92-0437, Jan. 1992.
5. Meakin, R.L., "Object X-Rays for Cutting Holes in Composite Overset Structured Grids," AIAA-2001-2537, June 2001.
6. Meakin, R.L., "Automatic Off-Body Grid Generation for Domains of Arbitrary Size," AIAA-2001-2536, June 2001.
7. Chan, W.M., Meakin, R.L., and Potsdam, M.A., "CHSSI Software for Geometrically Complex Unsteady Aerodynamic Applications," AIAA-2001-0539, Jan. 2001.
8. Meakin, R.L., "An Efficient Means of Adaptive Refinement within Systems of Overset Grids," AIAA-95-1722, June 1995.
9. Meakin, R.L., "On Adaptive Refinement and Overset Structured Grids," AIAA-97-1858, June 1997.
10. Benek, J.A., Buning, P.G., and Steger, J.L., "A 3-D CHIMERA Grid Embedding Technique," AIAA-85-1523, July 1985.
11. Pulliam, T.H., and Chaussee, D.S., "A Diagonalized Form of an Implicit Approximate Factorization Algorithm," *J. Comp. Physics*, Vol. 39, 1981.
12. Kandula, M., and Buning, P.G., "Implementation of LU-SGS Algorithm and Roe Upwinding Scheme in Overflow Thin-Layer Navier-Stokes Code," AIAA-94-2357, June 1994.
13. Jespersen, D.C., Pulliam, T.H., and Buning, P.G., "Recent Enhancements to OVERFLOW (Navier-Stokes Code)," AIAA-97-0644, Jan. 1997.
14. Jespersen, D.C., "Parallelism and OVERFLOW," NAS Technical Report NAS-98-013, NASA Ames Research Center, Moffett Field, CA, Oct. 1998.
15. Liou, M.S., and Buning, P.G., "Contribution of the Recent AUSM Schemes to the OVERFLOW Code: Implementation and Validation," AIAA-2000-4404, June 2000.
16. Pandya, S.A., Venkateswaran, S., and Pulliam, T.H., "Implementation of Preconditioned Dual-Time Procedures in OVERFLOW", AIAA-2003-0072, Jan. 2003.
17. Nichols, R., Tramel, R., and Buning, P., "Solver and Turbulence Model Upgrades to OVERFLOW 2 for unsteady and High-Speed Applications," AIAA-2006-2824, June 2006.
18. Nichols, R., Tramel, R., and Buning, P., "Evaluation of Two Higher Order WENO Schemes," AIAA-2007-3920, July 2007.
19. Beam, R., and Warming, R.F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form," *J. Comp. Physics*, Vol. 22, Sep. 1976, pp 87-110.
20. Klopfer, G.H., Hung, C.M., Van der Wijngaart, R.F., and Onufer, J.T., "A Diagonalized Diagonal Dominant Alternating Direction Implicit (D3ADI) Scheme and Subiteration Correction," AIAA-98-2824, June 1998.
21. Steger, J.L., and Warming, R.F., "Flux-Vector Splitting of the Inviscid Gasdynamics Equations with Application to Finite Difference Methods," *J. Comp. Physics*, Vol. 40, No. 2, Apr. 1981, pp 263-293.